MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2018/19

ECE3196 EMBEDDED SYSTEM DESIGN
(LE, EE, CE, TE)

13 MARCH 2019
9:00 a.m. – 11:00 a.m.
(2 Hours)

INSTRUCTIONS TO STUDENT

1. This Question paper consists of 9 pages including cover page and appendices with 4 Questions only.

2. Attempt **ALL** questions. All questions carry equal marks and the distribution of the marks for each question is given.

3. Please print all your answers in the Answer Booklet provided.

Question 1

(a) A smartwatch is a wearable embedded system in the form of a wristwatch. A typical smartwatch is equipped with a touchscreen, OLED display, accelerometer, gyroscope, heart rate sensor, vibration motor, buzzer and GPS receiver.

i. Describe what an embedded system is. [2 marks]

ii. Draw a block diagram to show the functional units and their relationship inside an embedded system for a typical smartwatch. [6 marks]

iii. Identify THREE (3) embedded system characteristics that are associated with a typical smartwatch. [3 marks]

(b) A disk subsystem has the following components and mean time to failure (MTTF):

- 10 disks, each rated at 1,000,000-hour MTTF
- 1 ATA controller, 500,000-hour MTTF
- 1 power supply, 200,000-hour MTTF
- 1 fan, 200,000-hour MTTF
- 1 ATA cable, 1,000,000-hour MTTF

Assuming that the lifetimes are exponentially distributed and that failures are independent:

i. Compute the MTTF of the system as a whole. [6 marks]

ii. If the mean time to repair (MTTR) the system is three days (72 hours), determine the availability of the system. [2 marks]

(c) Figure Q1(c) shows a hardware system that has the following assumptions on its reliability:

- Fault latency is zero.
- Any failure is permanent and any faulty processor is immediately identified and disconnected from the system, never repaired and reconnected.
- Failures are all independent.
- Let $R_i(t)$ be the reliability of component C_i over time interval $[0, t]$, where $t > 0$.

Suppose the reliability values for each component are given as:

$$R_1 = 0.75, R_3 = 0.8, R_4 = 0.75, R_5 = 0.8, R_6 = 0.8$$

If the reliability of the system is 0.5856, determine the reliability of component C_2 (R_2). List down all the necessary working steps clearly. [6 marks]

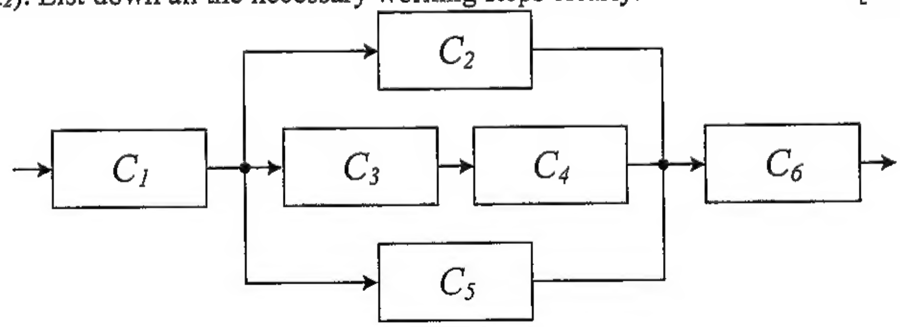

Figure Q1(c)

Continued ...

Question 2

(a) Processor technology is referring to the architecture of the computation engine used to implement a system's desired functionality:

i. List THREE (3) processor technologies and their benefits. [6 marks]

ii. Selection of the best processor must be aligned to the application. List THREE (3) efficiencies that must be considered during the selection process and identify ONE (1) method to improve each of these efficiencies. [3 marks]

(b) A computer has a processor with two L1 caches, one for instruction and one for data, and a L2 cache. Let τ be the access time for the two L1 caches. The miss penalties are approximately 10τ for transferring a block from L2 to L1, and 80τ for transferring a block from main memory to L2. Assume that the hit rates are the same for instructions and data and that the hit rates in the L1 and L2 caches are 0.95 and 0.85, respectively.

i. Determine the fraction of the accesses miss in both L1 and L2 caches, which require access to main memory. [2 marks]

ii. Compute the average access time as seen by the processor. [4 marks]

(c) Narrowband Internet of Things (NB-IoT) is a low-power wide-area network (LPWAN) technology developed by 3GPP to enable a wide range of new IoT devices and services. Briefly discuss TWO (2) strengths and TWO (2) weaknesses of NB-IoT comparing to LoRa technology. [6 marks]

(d) For the digital-to-analog converter (DAC) shown in Figure Q2(d), determine R_l if V_{ref} = 5V and V_{out} = -3.125V. [4 marks]

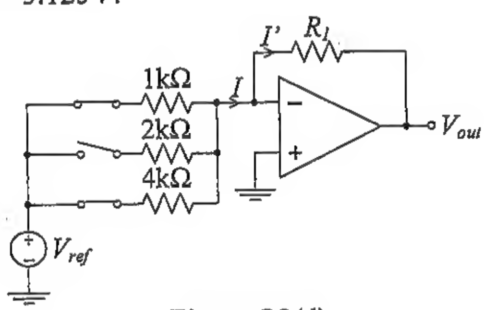

Figure Q2(d)

Continued...

Question 3

(a) In a multitasking system, a task can sometimes be referred to as a thread.

i. Draw a diagram to illustrate the relationship between FOUR (4) different states of a thread in mbed API. [4 marks]

ii. Briefly discuss any TWO (2) of the states illustrated in Q3(a)(i). [4 marks]

(b) Write an embedded C program with mbed SDK for the FRDM-KL25Z development board to get the status of the switches (D2 is LSB, D5 is MSB) and display the equivalent decimal digit on the 7-segment display as shown in the Figure Q3(b). For example, when none of the switches is pressed, the 7-segment display should show '0'; when the switches connected to D2 and D4 are pressed, the 7-segment display should show '5'. Refer to Table Q3(b) for the 7-segment codes. [8 marks]

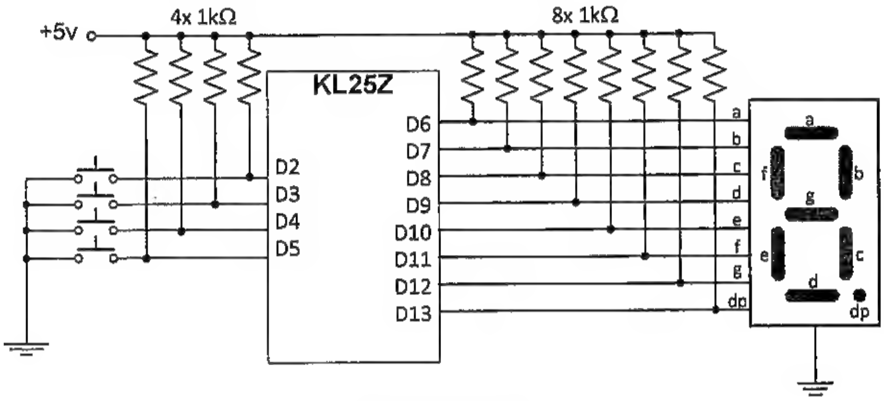

Figure Q3(b)

Digit	dp	g	f	e	d	c	b	a
0	0	0	1	1	1	1	1	1
1	0	0	0	0	0	1	1	0
2	0	1	0	1	1	0	1	1
3	0	1	0	0	1	1	1	1
4	0	1	1	0	0	1	1	0
5	0	1	1	0	1	1	0	1
6	0	1	1	1	1	1	0	0
7	0	0	0	0	0	1	1	1
8	0	1	1	1	1	1	1	1
9	0	1	1	0	1	1	1	1

Table Q3(b)

(c) As a trainee in an embedded system company, you are required to assist in developing an embedded C program using mbed SDK to control three LEDs via the FRDM-KL25Z development board. LED1 should blink at a rate of 4Hz, LED2 should blink at a rate of 2Hz and LED3 should blink at a rate of 1Hz. The operation starts once the board is powered and it will run continuously until power is removed. The LEDs are turned on by a logical '0' and they are connected to D2 (LED1), D3 (LED2) and D4 (LED3). Develop an embedded C program to perform the required operations in a multithreading manner by using mbed API for thread.

[9 marks]

Continued...

Question 4

(a) Convert the graphical Specification and Description Language (SDL) in Figure Q4(a) into the corresponding state diagram. [8 marks]

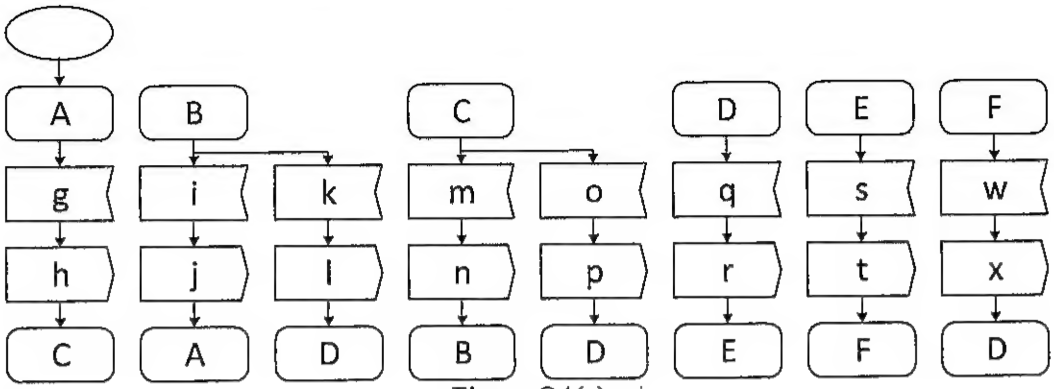

Figure Q4(a)

(b) The definition of a place/transition net is given as the following:
(P, T, F, K, W, M_0) is called a place/transition net if,

- ✓ $N = (P,T,F)$ is a net with places $p \in P$ and transitions $t \in T$.
- ✓ $K: P \rightarrow (\mathbb{N}_0 \cup \{\omega\}) \setminus \{0\}$ denotes the capacity of places (ω symbolizes infinite capacity).
- ✓ $W: F \rightarrow (\mathbb{N}_0 \setminus \{0\})$ denotes the weight of graph edges.
- ✓ $M_0: P \rightarrow \mathbb{N}_0 \cup \{\omega\}$ represents the initial marking of places.
- ✓ If $(p, t) \in F$ for a transition t and a place p, then p is an input place of t.
- ✓ If $(t, p) \in F$ for a transition t and a place p, then p is an output place of t.

i. Draw the initial place/transition net given the following features:
$P = \{a, b, c, d, e\}$,
$T = \{x, y\}$,
$F = \{(a, x), (b, x), (x, c), (x, d), (c, y), (d, y), (y, e)\}$,
$W = \{(a, x) \rightarrow 2, (b, x) \rightarrow 1, (x, c) \rightarrow 1, (x, d) \rightarrow 2, (c, y) \rightarrow 1, (d, y) \rightarrow 2, (y, e) \rightarrow 3\}$,
$M_0 = \{a \rightarrow 6, b \rightarrow 3, c \rightarrow 0, d \rightarrow 0, e \rightarrow 0\}$.
[6 marks]

ii. Redraw the place/transition net from Q4(b)(i) after one transition has taken place. [2 marks]

Continued...

(c) VHSIC Hardware Description Language (VHDL) is used to describe the structure of a digital circuit.

i. Write a VHDL entity for the decoder shown in Figure Q4(c)(i).

[3 marks]

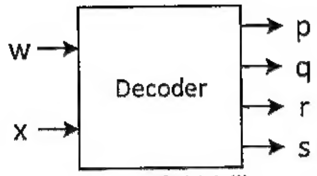

Figure Q4(c)(i)

ii. Write a VHDL structural body for the decoder circuit shown in Figure Q4(c)(ii).

[6 marks]

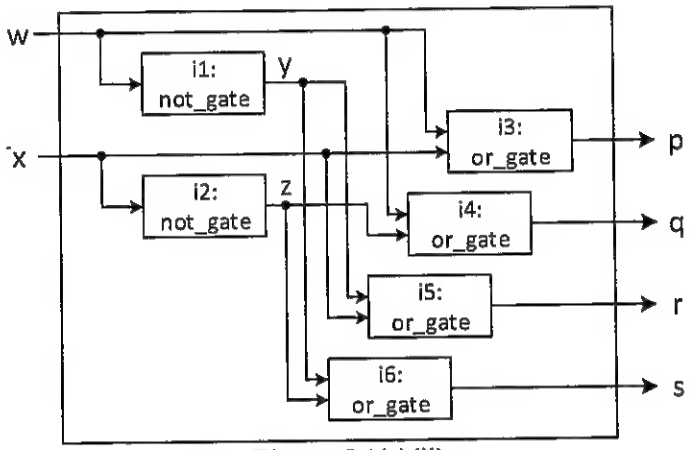

Figure Q4(c)(ii)

End of Paper

Appendix A: Selected mbed SDK functions

Function name	Description
Analog input	
AnalogIn (PinName pin)	Create an AnalogIn , connected to the specified pin
float read ()	Read the input voltage, represented as a float in the range [0.0, 1.0]
unsigned short read_u16 ()	Read the input voltage, represented as an unsigned short in the range [0x0, 0xFFFF]
operator float ()	An operator shorthand for read()
Analog output	
AnalogOut operator= (float percent)	An operator shorthand for write()
AnalogOut operator float ()	An operator shorthand for read()
AnalogOut (PinName pin)	Create an AnalogOut connected to the specified pin.
Digital Input	
DigitalIn (PinName pin)	Create a DigitalIn connected to the specified pin
DigitalIn (PinName pin, PinMode mode)	Create a DigitalIn connected to the specified pin and specify the input pin mode (PullUP, PullDown, OpenDrain)
int read ()	Read the input, represented as 0 or 1
mode (PinMode pull)	Set the input pin mode (PullUP, PullDown, OpenDrain)
BusIn (PinName pins)	Create an BusIn , connected to the specified pins.
Digital Output	
DigitalOut (PinName pin)	Create a DigitalOut connected to the specified pin
DigitalOut (PinName pin, int value)	Create a DigitalOut connected to the specified pin with an initial value (0 or 1)
write (int value)	Set the output, specified as 0 or 1
int read()	Return the output setting, represented as 0 or 1
BusOut (PinName pins)	Create an BusOut , connected to the specified pins.
Timer functions	
void start ()	Start the timer
void stop ()	Stop the timer
void reset ()	Reset the timer to 0
float read ()	Get the time passed in seconds
int read_ms ()	Get the time passed in milliseconds
int read_us ()	Get the time passed in microseconds
Time Ticker functions	
void attach (void(*fptr)(void), float t)	Attach a function to be called by the Ticker , specifying the interval in seconds
void attach (T *tptr, void(T::*mptr)(void), float t)	Attach a member function to be called by the Ticker , specifying the interval in seconds
void attach_us (void(*fptr)(void), unsigned int t)	Attach a function to be called by the Ticker , specifying the interval in micro-seconds
void attach_us (T *tptr, void(T::*mptr)(void), unsigned int t)	Attach a member function to be called by the Ticker , specifying the interval in micro-seconds
void detach ()	Detach the function.
static void irq (uint32_t id)	The handler registered with the underlying timer interrupt.

PWM output functions	
PwmOut (PinName pin)	Create a PwmOut connected to the specified pin.
Void write (float value)	Set the ouput duty-cycle, specified as a percentage (float)
float read ()	Return the current output duty-cycle setting, measured as a percentage (float)
void period (float seconds)	Set the PWM period, specified in seconds (float), keeping the duty cycle the same
void period_ms (int ms)	Set the PWM period, specified in milli-seconds (int), keeping the duty cycle the same
void period_us (int us)	Set the PWM period, specified in micro-seconds (int), keeping the duty cycle the same
void pulsewidth (float seconds)	Set the PWM pulsewidth, specified in seconds (float), keeping the period the same
void pulsewidth_ms (int ms)	Set the PWM pulsewidth, specified in milli-seconds (int), keeping the period the same
void pulsewidth_us (int us)	Set the PWM pulsewidth, specified in micro-seconds (int), keeping the period the same
PwmOut & operator= (float value)	A operator shorthand for write()
PwmOut & operator float ()	An operator shorthand for read()
UART functions	
Serial (PinName tx, PinName rx, const char *name=NULL)	Create a Serial port, connected to the specified transmit and receive pins
void baud (int baudrate)	Set the baud rate of the serial port
void format (int bits=8, Parity parity=SerialBase::None, int stop_bits=1)	Set the transmission format used by the serial port
int readable ()	Determine if there is a character available to read
int writeable ()	Determine if there is space available to write a character
void attach (void(*fptr)(void), IrqType type=RxIrq)	Attach a function to call whenever a serial interrupt is generated
void send_break ()	Generate a break condition on the serial line
void set_flow_control (Flow type, PinName flow1=NC, PinName flow2=NC)	Set the flow control type on the serial port
int putc(int ch, FILE *stream)	Writes the character ch to stream. Function returns the character written, or EOF if an error happens
int getc(FILE *stream)	Read a character from the stream, an EOF indicates the end of file is reached
int printf(const char *format, ...)	Prints output both text string and data, according to format and other arguments passed to printf()
Other functions	
void wait (float s)	Wait for a number of seconds
void wait_ms (int ms)	Wait for a number of milliseconds
void wait_us (int us)	Wait for a number of microseconds

Appendix B: Selected mbed API for Thread

mbed API for Thread	
Thread (osPriority priority=osPriorityNormal, uint32_t stack_size=DEFAULT_STACK_SIZE, unsigned char *stack_pointer=NULL)	Allocate a new thread without starting execution.
osStatus start(task)	Create a new thread, and start it to execute the specified task/function
osStatus terminate ()	Terminate execution of a thread and remove it from Active Threads
osStatus set_priority (osPriority priority)	Set priority of an active thread
osPriority get_priority ()	Get priority of an active thread
int32_t signal_set (int32_t signals)	Set the specified Signal Flags of an active thread
State get_state ()	State of this Thread
Mutex ()	Create and Initialize a Mutex object
osStatus lock (uint32_t millisec=osWaitForever)	Wait until a Mutex becomes available.
osStatus unlock ()	Unlock the mutex that has previously been locked by the same thread
osStatus Thread::wait(uint32_t ms)	Wait for a number of milliseconds